

PLP - Segundo Recuperatorio - 2^{do} cuatrimestre de 2021

Este examen se aprueba obteniendo al menos dos ejercicios bien menos (B-) y uno regular (R). Las notas para cada ejercicio son: -, I, R, B-, B. Poner nombre, apellido, número de orden y cantidad de hojas en la primera hoja, y numerar las hojas. Se puede utilizar todo lo definido en las prácticas y todo lo que se dio en clase, colocando referencias claras.

El orden de los ejercicios es arbitrario. Recomendamos leer el parcial completo antes de empezar a resolver.

Ejercicio 1 - Lenguajes de Programación

a) Sean las siguientes funciones definidas en Haskell:

```
producto xs ys = [(x, y) | x <- xs, y <- ys]

entrelazar = foldr (\x rec ys->if null ys then x:recys else x:head ys:rec (tail ys)) id

ultimos = foldl (\rec x n->if n == 0 then [] else rec (n-1) ++ [x]) (const [])
```

Se espera que, al aplicarlas a los argumentos correspondientes, estas funciones arrojen respectivamente el producto cartesiano de las listas recibidas, las listas entrelazadas y los últimos n elementos de una lista. Explicar, para cada una de ellas, si funciona correctamente al recibir listas infinitas y/o finitas.

b) Sea el siguiente código JavaScript:

```
function Complex(r, i) {
  this.real = r;
  this.imag = i;
}

let a {real: 1, imag: 1};
let b = Object.create(a);
b.real = 5;
let c = new Complex(2,1);
let d = Object.assign({}, c);

Complex.prototype.suma = function(otro){
  let res = Object.create(this);
  res.real += otro.real;
  res.imag += otro.imag;
  return res;
}

let e = c.suma(a);
let f = Object.assign({}, e);
Object.setPrototypeOf(f, Object.getPrototypeOf(a));
Object.setPrototypeOf(a, Complex.prototype);
Object.setPrototypeOf(b, Object.getPrototypeOf(Complex));
```

Indicar, para cada uno de los objetos a, b, c, d, e, f, cuál es su prototipo luego de ejecutar el código, y qué atributos conoce cada objeto, ya sea por tenerlos como atributos propios o por herencia. (Considerar solamente los atributos real, imag y suma.)

c) Se definieron los siguientes predicados en Prolog para analizar los puntajes de distintos equipos en un torneo:

`puntaje(?Equipo, ?P)`, que tiene éxito cuando P es el puntaje obtenido por `Equipo` en el torneo.

`vencio(?E1, ?E2)`, que es verdadero cuando el equipo $E1$ venció a $E2$ en un encuentro del torneo.

Se definieron además los siguientes predicados:

```
g(Equipo) :- puntaje(Equipo, P), not((puntaje(E2, P2), P2 > P)).
```

```
h(E1, E2) :- puntaje(E2, P2), vencio(E3, E1), puntaje(E3, P3), P3 < P2.
```

```
i(E1, E2) :- vencio(E2, E1), not(h(E1, E2)).
```

¿Qué representan los predicados $g/2$ e $i/1$? ¿Hay algún caso en el que puedan devolver más de una solución? ¿Cuáles son los patrones de instanciación más apropiados para ellos? Justificar.

Ejercicio 2 - Cálculo Sigma

a) Sean las siguientes clases:

$$plantaClass \stackrel{\text{def}}{=} [\text{new} = \varsigma(c)[\text{altura} = c.\text{altura}, \text{crecer} = \varsigma(t)c.\text{crecer}(t)], \\ \text{altura} = 10, \\ \text{crecer} = \varsigma(c)\lambda(t)t.\text{altura} := (t.\text{altura} + 10)]$$

$$frutalClass \stackrel{\text{def}}{=} [\text{new} = \varsigma(c)[\text{altura} = c.\text{altura}, \text{crecer} = \varsigma(t)c.\text{crecer}(t), \text{cantFrutos} = \varsigma(t)c.\text{cantFrutos}(t)], \\ \text{altura} = plantaClass.\text{altura}, \\ \text{crecer} = \lambda(t)plantaClass.\text{crecer}(t).\text{cantFrutos} := t.\text{cantFrutos} + 1, \\ \text{cantFrutos} = 0]$$

Mostrar cómo evalúa $frutalClass.\text{new}.\text{crecer}$.

Suponer que los números son valores y sus sumas devuelven los valores esperados.

b) Definir en el cálculo de objetos, el objeto **nuevoDicc** que representa un diccionario sin claves definidas y sabe responder los siguientes mensajes:

- **def?(x)**, que devuelve **true** si la clave **x** está definida en el diccionario.
- **definir(x)(s)**, que define la clave **x** con significado **s** (si ya estaba definida, el nuevo valor reemplaza al anterior).
- **obtener(x)**, que devuelve el significado asociado a la clave **x**. Si la clave no está definida, no devuelve un valor, sino que se cuelga.

Suponer definida la operación **==** para las claves.

Ejercicio 3 - Resolución Lógica

a) Convertir las siguientes fórmulas a Forma Clausal:

1. $\forall X \text{ racional}(X) \supset \text{racional}(X/2)$
2. $\forall X \text{ mayor}(X, 0) \supset \text{mayor}(X/2, 0)$
3. $\forall X \text{ mayor}(X, 0) \supset \text{mayor}(X, X/2)$

Donde $/2$ es una función. Si la notación confunde, se puede escribir $f(X)$ en lugar de $X/2$.

b) Utilizar el método de resolución para probar que no existe un mínimo número racional mayor que 0. Es decir:

$$\neg \exists R(\text{racional}(R) \wedge \text{mayor}(R, 0) \wedge \neg \exists Q(\text{racional}(Q) \wedge \text{mayor}(Q, 0) \wedge \text{mayor}(R, Q)))$$

No está permitido usar el resolutor.

c) El método de resolución utilizado en el punto b), ¿fue SLD? Justificar.